



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/715,316	11/17/2003	Ciprian Agapi	N0484.70570US00	5087
23628 7590 07/28/2009 WOLF GREENFIELD & SACKS, P.C. 600 ATLANTIC AVENUE BOSTON, MA 02210-2206				
EXAMINER				
LERNER, MARTIN				
ART UNIT		PAPER NUMBER		
2626				
MAIL DATE		DELIVERY MODE		
07/28/2009		PAPER		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/715,316

Applicant(s)

AGAPI ET AL.

Examiner

MARTIN LERNER

Art Unit

2626

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 24 June 2009.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1 to 10 and 30 to 39 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1 to 10 and 30 to 39 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/S508)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

2. Claims 1 to 10 and 30 to 39 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Marx et al.* in view of *Zirngibl et al.*

Concerning independent claims 1, 30, and 39, *Marx et al.* discloses a method, system, and computer readable medium for developing interactive speech applications, comprising:

“presenting a [style-]selection menu for a plurality of catch styles that allows for selection of one or more of the catch [styles], each catch [style] defining a system response to each of a plurality of catch events, wherein each catch [style] provides a different level of complexity with regard to preparing a system’s audio response to be played in a dialog turn, and the plurality of catch events comprises an event selected from the group consisting of a user request for help, a non-input entry, and a non-matching entry” – Dialogue Module templates are provided as pre-packaged modules that can be used to create applications that have internally consistent software code (column 4, lines 33 to 36); dialogue modules are stored as graphically represented icons in a graphical display, in which icons for the subset of dialogue modules are

selected in the graphical display in response to user input; the interactive speech application is generated based upon the graphical representation (column 3, line 66 to column 4, line 15: Figure 7); a system comprises a plurality of Dialogue Modules, each designed for performing a specific dialogue task such as outputting a prompt, identifying the caller's speech as a recognized item of a predefined list, identifying a caller's speech as an affirmative or negative (Yes/No) response, or identifying strings of characters spelled by the caller (column 6, lines 42 to 48: Figure 4); by providing the interface, the Dialogue Modules 430 allow a developer to develop a Service 410 without a detailed understanding of the Speech Components, 440, 450, whose functions include outputting prompts to callers and receiving and processing input speech from callers (column 6, line 64 to column 7, line 3: Figure 4); Figure 7 shows how Dialogue Module templates 730 are selected from a list of on-screen icons, which is equivalent to "presenting a . . . selection menu . . . that allows for selection"; each Dialogue Module performs a discrete task, and includes a value indicating its termination condition; termination conditions include SUCCESS, indicating a successful completion of a dialogue task, TIMEOUT, indicating that the caller did not respond within a predetermined timeout period, and ERROR, indicating that the system could not recognize the caller's response (column 8, lines 19 to 31); thus, broadly, a "catch event" corresponds to a termination condition of a TIMEOUT ("a non-input entry") or ERROR ("a non-matching entry"), where the system could not recognize the user response within a predetermined timeout period ("an event being selected from the group consisting of . . . a non-input entry, and a non-matching entry"); Dialogue Module

templates include error recovery methods when the Service does not collect a response from the caller during the timeout period; at least three “styles” of default error recovery procedures are disclosed: (1) retry by the same method, where a user is prompted again with the same prompt for a maximum number of times, (2) an apology prompt method, where a user is prompted with an apology, and prompted to repeat an answer now, and (3) a fallback method where a user is requested to spell a response or enter through touch-tone keys (column 13, lines 10 to 67: Figure 6: Steps 640, 650a, and 660); Dialogue Modules are provided in a Baseline Configuration library of default settings, including standard parameters, which can be customized (column 17, lines 5 to 34: Figure 8); a retry method with a reprompt represents a relatively simple audio response, of repeating, “Please say your answer now”, while a fallback method is a relatively more complex request for a user to spell his or her response (column 13, lines 23 to 67: Figure 6: Steps 640a and 640b); similarly, Error Recovery options include relatively simple default general reprompt prompts in Baseline 820 and System 830 libraries, or relatively complex options where a developer can customize text in a prompt for specific instances of a Dialogue Module by selecting options or by providing the text as shown in Figure 16 (“wherein each catch style provides a different level of complexity with regard to preparing a system’s audio response to be played in a dialog turn”) (column 20, line 16 to column 21, line 8: Figures 8 and 16); thus, audio prompts are provided with at least two degrees of complexity;

“upon selection of a catch [style], preparing the system’s audio response for each catch event” – selecting an error recovery option allows a developer to customize the

error recovery parameters within a Dialogue Module instance (column 20, lines 15 to 21: Figure 9); whether a developer selects a Dialogue Module with default parameters, or customizes a Dialogue Module, each configuration parameter causes a change in operation of the dialogue module when the interactive speech program executes (Abstract); implicitly, then, the interactive speech program "prepares the system response" in accordance with the parameters specified by the developer for each error condition ("catch").

Concerning independent claim 1, 30, and 39, the only elements not expressly disclosed by *Marx et al.* are the concepts of "style"-selection and "catch styles". *Marx et al.* discloses a plurality of default templates for error conditions when a user response is not understood, where an error condition is equivalent to a "catch", but omits the concept of a "style" in describing a "catch" and a process of selection. However, it is known in the art of voice services to provide style sheets to create interactive voice services. Specifically, *Zirngibl et al.* teaches a system and method for creation and automatic deployment of personalized dynamic and interactive voice services, where XML (extensible style sheet language) style sheets are provided to create voice services. An objective is to maximum an administrator's voice service building capability. (Column 11, Lines 32 to 49) It would have been obvious to one having ordinary skill in the art to apply a concept of "style" to selection of "catch styles" as taught by *Zirngibl et al.* in a Dialogue Module selection method of *Marx et al.* for a purpose of maximizing an administrator's voice service building capability.

Concerning claims 2 and 31, *Marx et al.* discloses that a Dialogue Module may be customized by a developer to include content of prompts ("a new audio message to be played in response to a particular catch event") (column 20, lines 28 to 34: Figure 16); in one embodiment, a prompt can be specified by a filename, but a prompt may be specified by inputting text ("presenting one or more text fields for receiving a contextual message, the contextual message entered in each text field") if a text-to-speech synthesizer is used (column 18, lines 30 to 40; column 20, lines 58 to 63; column 21, lines 5 to 8).

Concerning claims 3 to 4 and 32 to 33, *Marx et al.* discloses that Dialogue Module templates may have a default initial prompt, but may require a custom initial prompt to be provided by a developer (column 18, lines 40 to 45); if a default prompt is used to an error condition, then the "contextual message is the same for each catch event"; however, if a prompt is customized for an error condition, then the "contextual message is different for each catch event".

Concerning claim 5, *Marx et al.* discloses that one of the Dialogue Module templates for error recovery involves replaying a prompt for a number of retries (column 13, lines 10 to 39: Figure 6: Step 640).

Concerning claims 6 and 34, *Marx et al.* discloses that Error Recovery 950 allows a developer to view and modify error recovery parameters (column 18, lines 1 to 3; column 20, lines 28 to 33: Figure 16).

Concerning claims 7 and 35, *Marx et al.* discloses an ItemList Module 520 can terminate on an ERROR condition 540, and take appropriate termination actions,

including to transfer the caller to a live operator (column 9, lines 62 to 65: Figure 5: Step 540); an ItemList Module lets a developer define allowable responses to a caller prompt and return a termination condition ("identifying a final action to be taken") (column 15, line 66 to column 16, line 9); Error Recovery 950 allows a developer to view and modify error recovery parameters (column 18, lines 1 to 3; column 20, lines 28 to 33: Figure 16).

Concerning claims 8 to 10 and 36 to 38, *Marx et al.* discloses that a developer can customize at least a "timeout" parameter that sets a predetermined time period for the caller to respond after the output of a prompt (column 11, lines 7 to 16); thus, at least customizing a "timeout" period corresponds to "inserting variables in a contextual message"; moreover, a "timeout" parameter defines "pauses of specific duration values" in a message after the prompt, and can "enable acceleration of a system timeout" because a shorter "timeout" period corresponds to an acceleration of an error recovery procedure.

Response to Arguments

3. Applicants' arguments filed 24 June 2009 have been fully considered but they are not persuasive.

Firstly, Applicants allege that the Office Action concedes that *Marx et al.* does not teach "style" selection or "catch style", but alleges that these features are taught by *Zimgibl et al.*

However, it is maintained that *Marx et al.* does disclose the features of "style" selection or "catch style", implicitly, and that the rejection only alleges that those features are not "expressly" disclosed by *Marx et al.* Although Applicants are permitted to be their own lexicographer by coining new terminology to describe their method, system, and computer program, that terminology must still be construed as broadly as the terms reasonably allow in accordance with their plain meaning. *In re American Academy of Science Tech Center*, 367 F.3d 1359, 1369, 70 USPQ2d 1827, 1834 (Fed. Cir. 2004) See MPEP §2111.01. Basically, it is maintained that *Marx et al.* is disclosing equivalents to "style selection" and "catch styles", although these features are not clearly and expressly disclosed. In one sense, the term "style" is defined as "a distinctive manner of expression". In another sense, the term "style" is simply defined as a "sort" or "type". *Marx et al.* clearly discloses a variety of "sorts" or "types" of Dialogue Modules 730. These dialogue modules are templates that can be customized to respond to a variety of "sorts" or "types" of error conditions. Similarly, the term "catch" for describing a "catch style" or "catch event" is a coined term that should be

broadly construed in accordance with plain meaning and what is actually disclosed by Applicants' Specification. Essentially, a "catch" is what an error condition is supposed to do. The Dialogue Module is designed to "catch" certain error conditions, so that the error condition is "caught" by the software. *Marx et al.* discloses dialogue module templates that are designed to "catch" error conditions, so that the error conditions are "caught" by the dialogue module software.

It is maintained, then, that the features of "style" selection, "catch styles", and "catch events" are all disclosed implicitly by *Marx et al.* *Zirngibl et al.* is cited cumulatively for the teaching that an XML style sheet is a template for programming. Indeed, *Zirngibl et al.* may not be strictly necessary for a rejection because one skilled in the art could see from the very definition of the term "style" that nothing more is being implied than a "sort" or "type". Still, though, *Zirngibl et al.* teaches that a "style" sheet is an element of programming that maximizes an administrator's voice service building ability. *Zirngibl et al.*, then, provides some supplemental teaching to identify a programming template with a "style" in accordance with an ordinary usage in the art of programming. Mainly, *Zirngibl et al.* is cited for assistance with a term of vocabulary.

Secondly, Applicants argue that the prior art does not disclose a "style-selection menu". This position is traversed.

Marx et al. pretty clearly discloses a "menu" in Figure 7. While one could quibble about what constitutes a "menu", *Marx et al.* shows a plurality of Dialogue Module templates 730 that can be dropped and dragged from a stencil palette window 710 on the left side of a GUI 700 to a main workspace 740 on the right side of the GUI 700.

(Column 16, Lines 26 to 53: Figure 7) A slider bar on the right side of the stencil palette window 710 permits a programmer to scroll down to find any one of a further plurality of Dialogue Module templates 730. (*Marx et al.* is disclosing that at least some of these Dialogue Modules are directed to dealing with error conditions.) It is unclear how Applicants might wish the term "menu" to be interpreted that would exclude the stencil palette window 710 from equivalently being called a "menu". Certainly, there are simpler forms of "menus" that involve only numbered (or lettered) lists of items that permit a user to select an item on a list by number (or letter), but a display of a plurality of icons in an array for selection by a programmer would qualify equivalently as a "menu", too. And dropping and dragging of icons from a stencil palette window 710 to a main workspace 740 so as to link a plurality of selected program states is simply equivalent to a "selection".

Moreover, the prior comments directed to the coined terms "style" and "catch" apply here, also. Thus, *Marx et al.* discloses not only a "menu", but, implicitly, "a selection menu" and "a style-selection menu" because there are a plurality of "styles", *i.e.* "sorts" or "types", of Dialogue Module templates that respond to error conditions. Similarly, *Marx et al.* is disclosing "a style-selection menu for a plurality of catch styles" because the Dialogue Module templates that respond to error conditions are designed to "catch" the error conditions, so that the circumstances of the error are dealt with, or "caught" – and, preferably ultimately corrected.

Therefore, the rejection of claims 1 to 10 and 30 to 39 under 35 U.S.C. §103(a) as being unpatentable over *Marx et al.* in view of *Zirngibl et al.* is proper.

Conclusion

4. **THIS ACTION IS MADE FINAL.** Applicants are reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to MARTIN LERNER whose telephone number is (571)272-7608. The examiner can normally be reached on 8:30 AM to 6:00 PM Monday to Thursday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, David R. Hudspeth can be reached on (571) 272-7843. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for

Art Unit: 2626

published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Martin Lerner/
Primary Examiner
Art Unit 2626
July 24, 2009